Frameworks: Comparison and Correspondence for Three Archetypes

Richard Martin

Tinwisle Corporation
Bloomington, Indiana

Edward Robertson

Computer Science Department
Indiana University

Outline - Frameworks: Comparison and Correspondence for Three Archetypes

- Sampler
- Foundation
- ISO/DIS 19439
- C4ISR Version 2.0
- Compare Features
- Correspondence

Architectural Representations

- FRAT pyramid
- ARC CMM sphere
- Rockwell-Collins cube
- PERA wind chime
- GERA tower
  (ISO 15704:2000 Annex A)

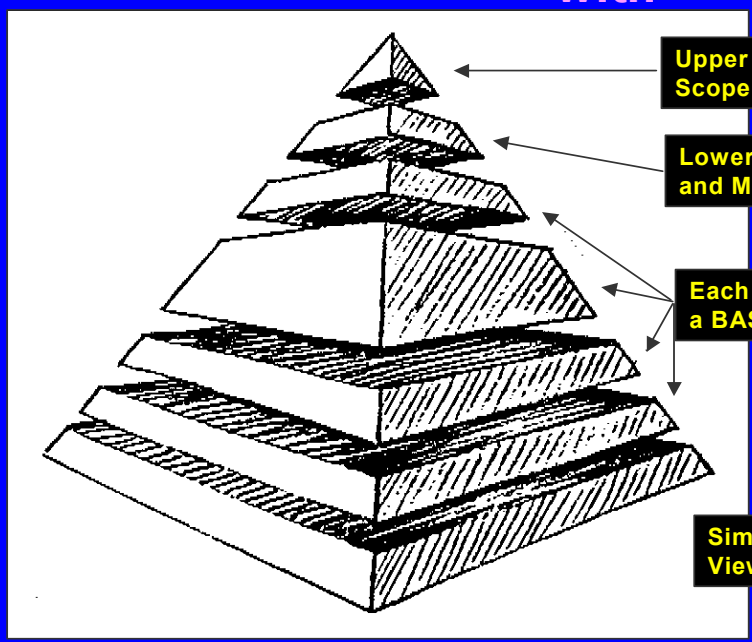
- ISO/DIS 19439:2002 tower
- Zachman grid
- C4ISR Version 2.0 triad

Presentation focus is on latter three
with reference to fifth.

FRAT

Function, Requirements, Answers, Test

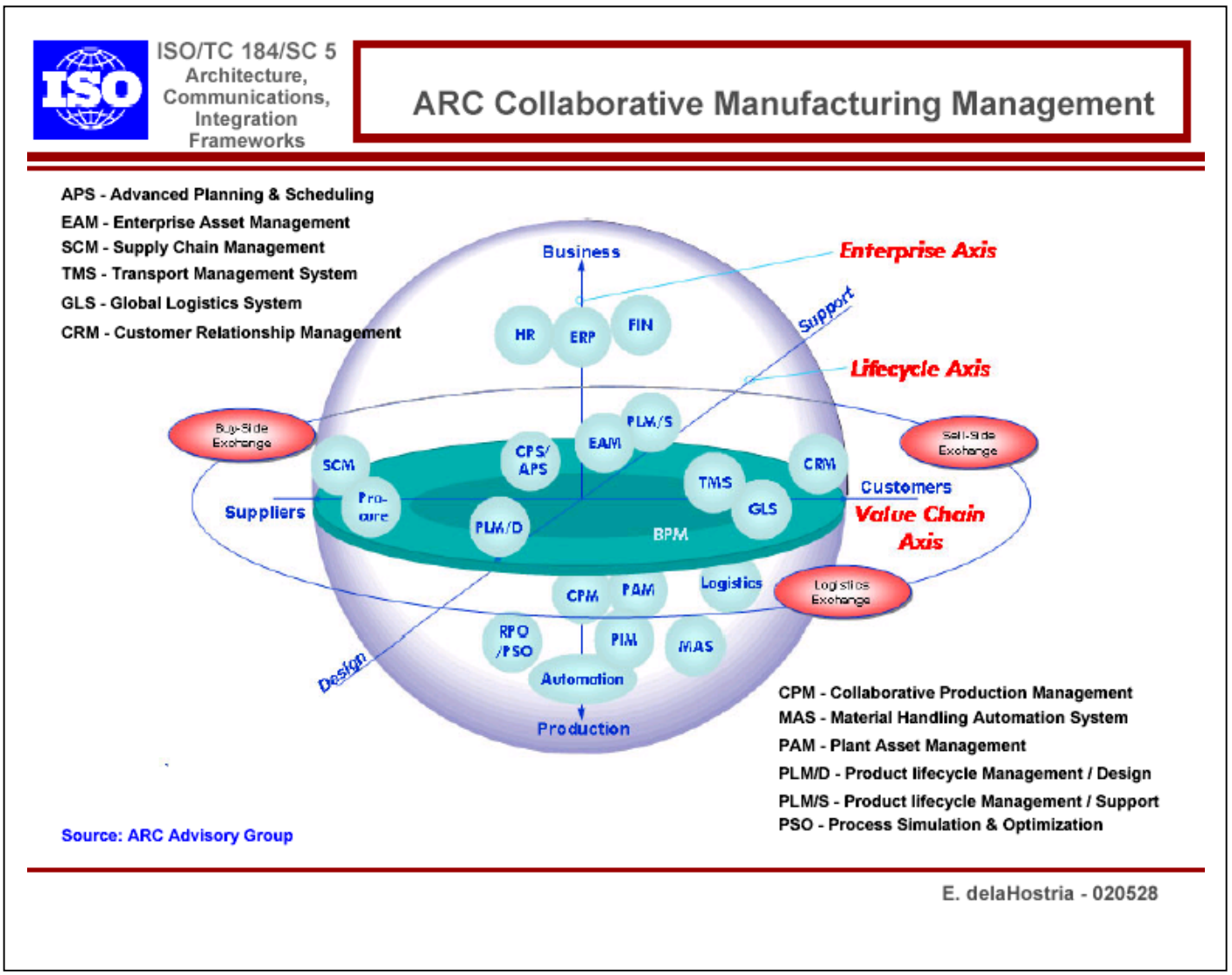
Systems thinking -
each of the FRAT views can be described with



Source: B. W. Mar, B. G. Morais, FRAT – A Basic Framework for Systems Engineering, INCOSE 2002

Detail elaboration adds both depth and breadth to the system description

ARC CMM

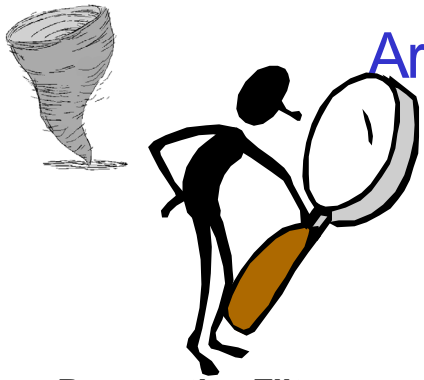


Source: ISO/TC 184/SC5 N913, E. delaHostria, Chairman, and ARC Advisory Group (used with permission)

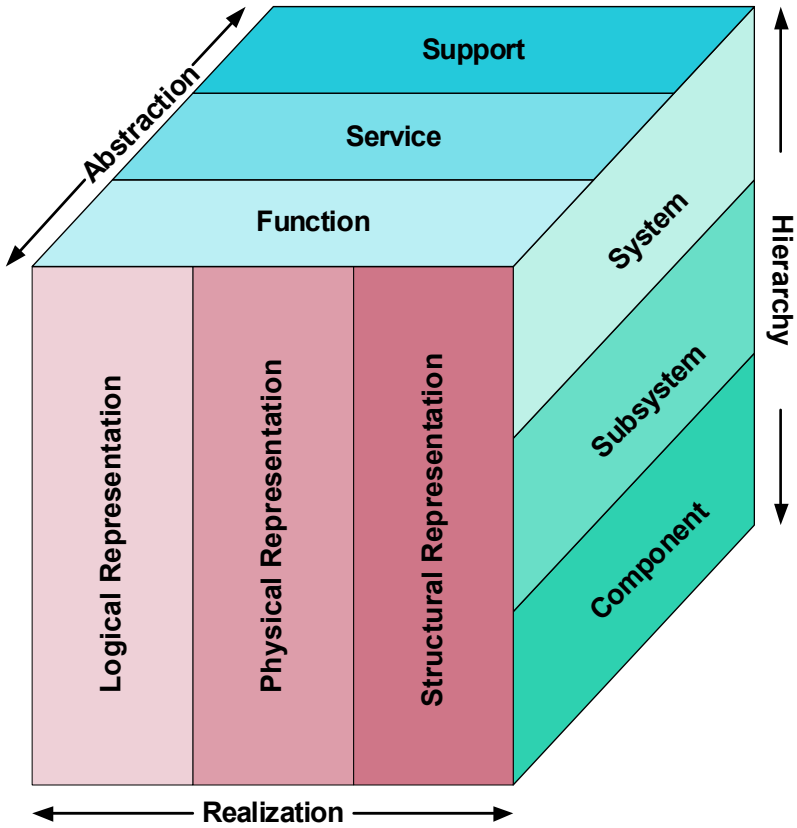
Aligning functional applications along axis to identify dimensions of the global manufacturing enterprise

Rockwell Collins

System Architectural Representation



Architectural Perspectives



Perspective Filters:

- Development & Verification
- Manufacturing & Production
- Storage & Transportation
- Installation & Deployment
- Simulation & Training
- Operational
- Maintenance & Support
- Disposal
- Project
- Safety
- Functional
- Physical
- Information/ Data Flow

6 Dec 2001
Ver 2.5

37

Source: R. W. Jorgensen, Architectural Abstractions, INCOSE 2002.

Purdue Enterprise Reference Architecture

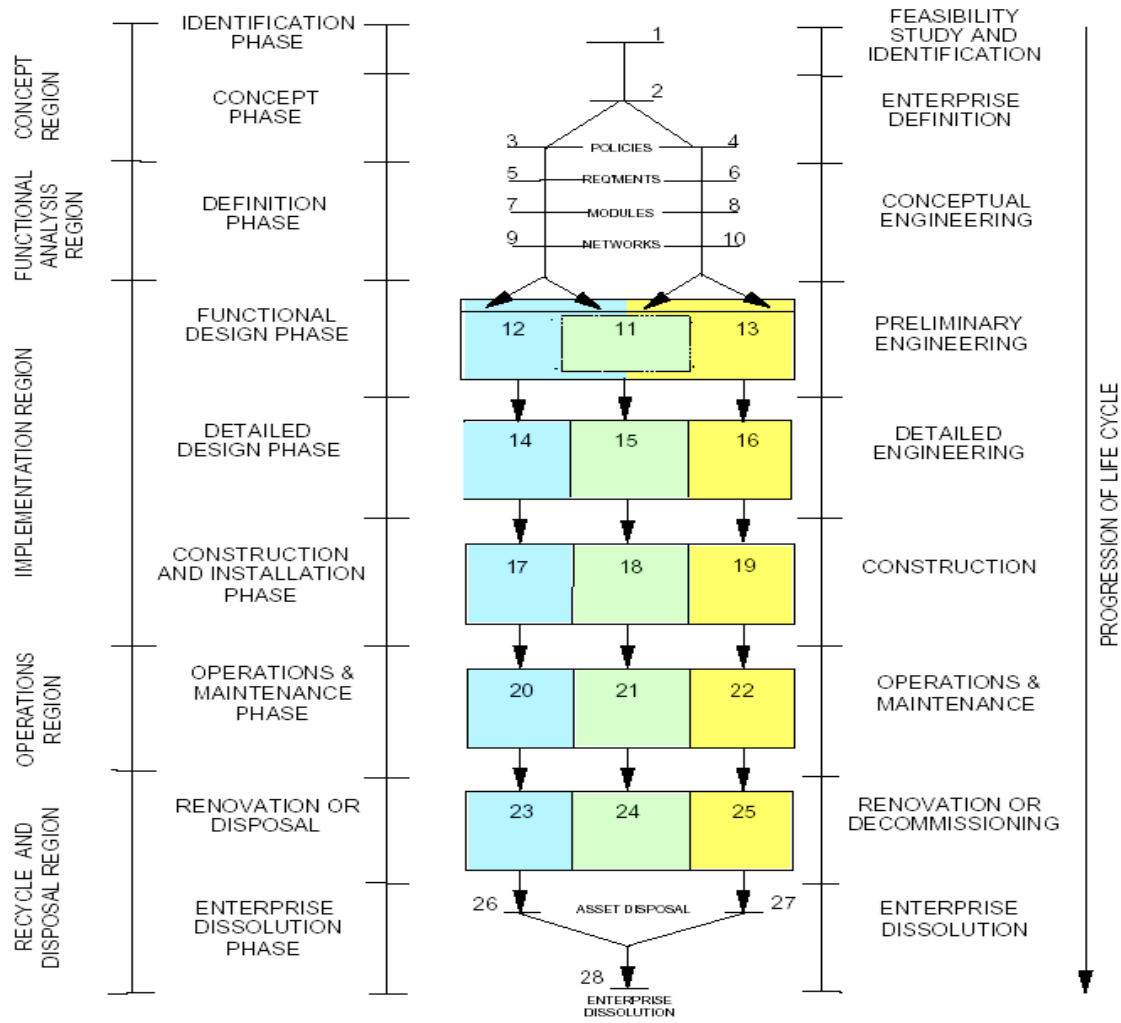
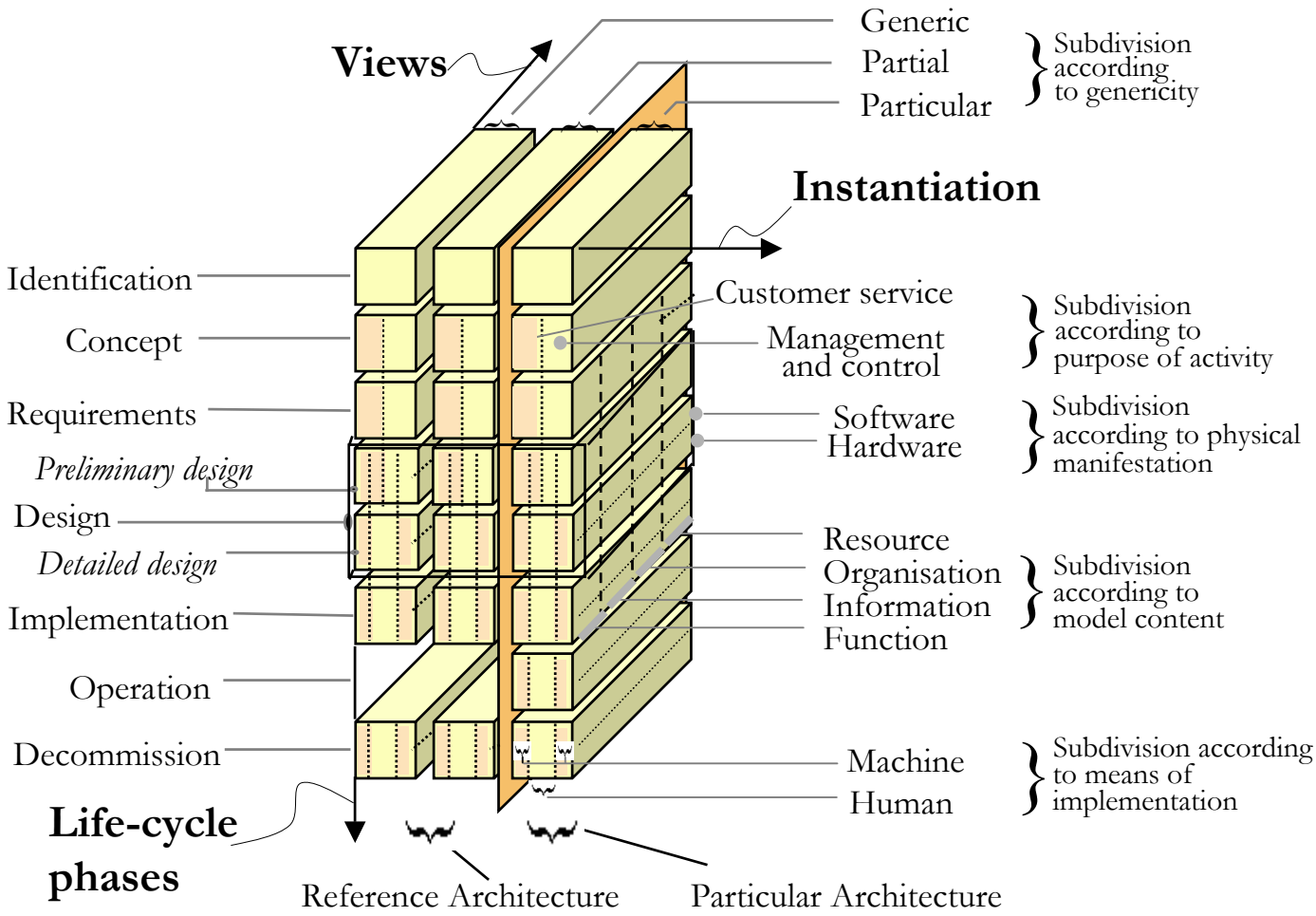


FIGURE 2 - OVERALL FORM OF THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE DIAGRAM SHOWING VARIOUS FORMS OF THE LIFE CYCLE

Source: T. J. Williams, A Handbook on Master Planning and Implementation for Enterprise Integration Programs, Institute for Interdisiplinary Engineering Studies, Purdue University.

GERAM V 1.6.3

Generalised Enterprise Reference Architecture and Methodology

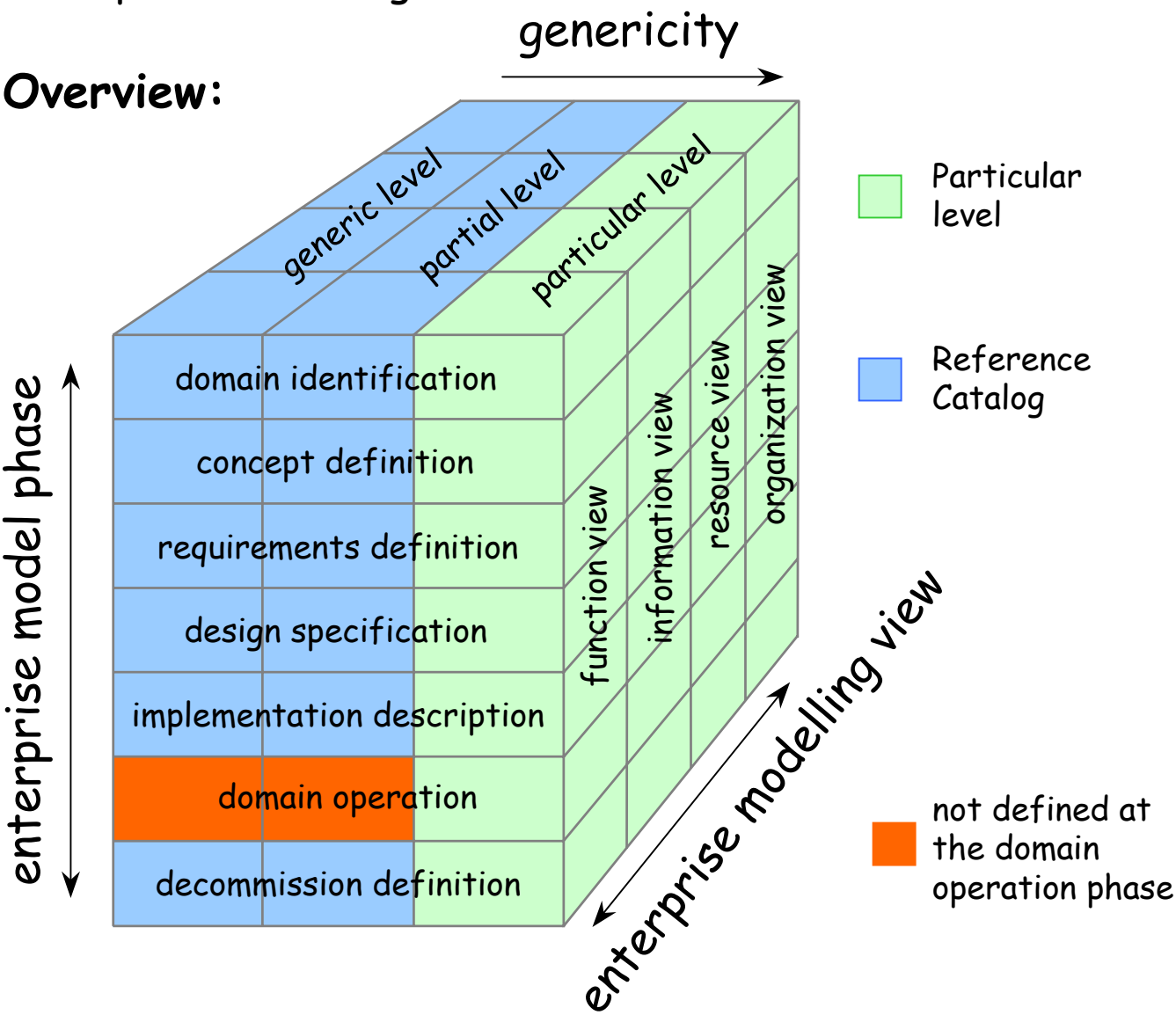


Source: ISO 15704:2000 Annex A and Figure 10, The GERA modelling Framework of GERAM [GERAM V1.6.3 http://www.cit.gu.edu.au/~bernus](used with permission)

ISO/DIS 19439

CIM Systems Integration: Framework for Enterprise Modelling

Overview:



Source: International Standards Organization ISO/CEN parallel enquiry draft prEN ISO 19439 of 4/22/2002

Zachman framework
(of excruciating detail)

ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Function = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Oreanizations Important to the Business People = Major Organizations	List of Events Significant to the Business Time = Major Business Event	List of Business Goals/Strat Ends/Means=Major Bus. Goal/ Critical Success Factor	SCOPE (CONTEXTUAL) *Planner*
ENTERPRISE MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Logistics Network Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	ENTERPRISE MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. "Application Architecture" Proc. = Application Function I/O = User Views	e.g. "Distributed System Architecture" Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. "System Design" Proc.= Computer Function I/O = Screen/Device Formats	e.g. "System Architecture" Node = Hardware/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY CONSTRAINED MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. "Program" Proc.= Language Stmt I/O = Control Block	e.g. "Network Architecture" Node = Addresses Link = Protocols	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

Zachman Institute for Framework Advancement - (810) 231-0531 Copyright - John A. Zachman, Zachman International

Role by **I**nterrogative grid of cells containing models of the enterprise

A proto-typical **F**ramework!

C4ISR Version 2.0

Architectural Views

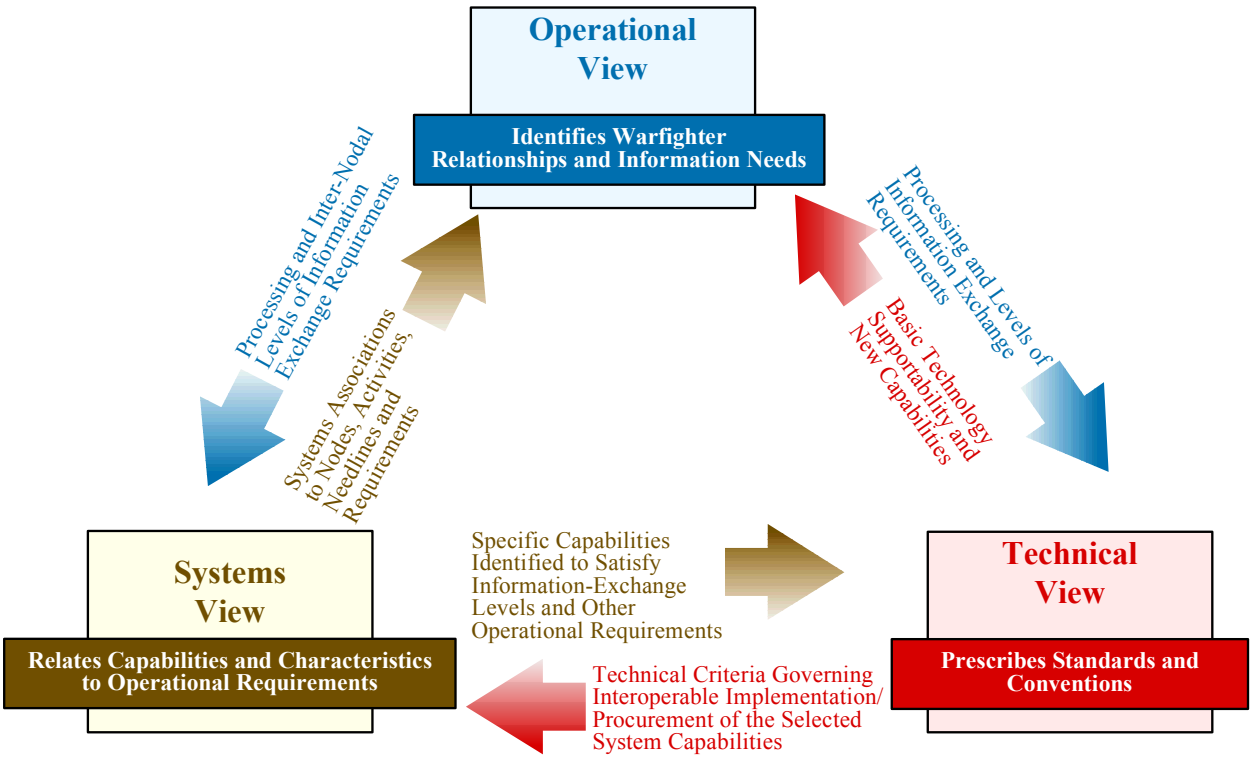


Figure 2-2. Fundamental Linkages Among the Views

"… intended to ensure that the architecture descriptions developed by the Commands, Services, and Agencies are interrelatable between and among each organization's operational, systems, and technical architecture views, and are comparable and integratable across Joint and combined organizational boundaries."

Source: Architecture Working Group, C4ISR Architecture Framework Version 2.0, 1997

General Modeling Principles
(including frameworks)

- Models are formal artifacts developed and used by people.
- Complexity tradeoff exists between modeling medium and model instances in that medium.
- Naming serves as the bridge between the formal and the human.
- Both grid (ordinant) and tree (decomposition) structures appear in models.
- Scale dimensions include:
  concept (abstract to concrete),
  scope (general to special) and
  detail (coarse to fine).
- Separate model and instance decompositions – do not confuse meta-levels.
- Separate model and instance constraints.
- Don't hide architecture in methodology.

Framework Principles

- A framework is a mechanism, not policy.
- Formalize the framework approach, not one particular framework.
- Interconnections should not be encoded in structure.
- Names have two uses: ordinant coordinate, and one member of collection.
- One dimension reflects the purposive nature of the framework and is usually ordered.
- Along the purposive dimension, all preceding material is relevant.
- Recursion is a structural mechanism, iteration is a process mechanism.
- Views make a massive model comprehensible.

Framework meta-meta model

Structure:

- both tree (decomposition) and grid (ordinant)
- frames and sub-frames

Connections:

- between frame components
- respects purposive order

Constraints:

- model and instance
- beyond structure and connection

Views:

- generalizes "view" in existing frameworks
- defined on structure
- attempts to carry forward connections and constraints

Formal framework model

branch frames:

$\mathcal{F}_\alpha \quad \langle IC_\alpha, OC_\alpha, \mathcal{SF}_\alpha, \Phi_\alpha \rangle$

leaf frames:

$\mathcal{F}_\alpha \quad \langle IC_\alpha, OC_\alpha, \quad S_\alpha \rangle$

where

$IC_\alpha \quad \subseteq \mathcal{D}$

$OC_\alpha \quad \subseteq \mathcal{D}$

$\left.\begin{matrix} \varepsilon OC_{\alpha,r} \\ \varepsilon IC_{\alpha,r} \end{matrix}\right\} \quad \subset \mathcal{D}$ restricted to row r

$\mathcal{SF}_\alpha \quad : \mathcal{R} \times I \times \mathcal{D} \rightarrow \mathcal{F} \cup \mathcal{VF}$

$\Phi_\alpha \quad \subseteq \cup_{r \in \{\theta\} \cup \mathcal{R}} (\varepsilon OC_{\alpha,r} \times \varepsilon \mathrm{I} C_{\alpha,r'})$

$\mathcal{Types} \quad \mathcal{D} \cup \{\text{SET OF } d' : d' \in \mathcal{D}\}$

$S_\alpha \quad : \mathcal{D} \rightarrow \cup_{n \in \aleph} \mathcal{Types}_\alpha{}^n$

Source: R. Martin & E. Robertson, Formalization of Multi-level Zachman Frameworks, 1999, http://www.cs.indiana.edu/Research/techreports/TR522.shtml

Entities in time –

The characterization of a framework entity with respect to time informs us about the nature of change in the framework's context.

continuant vs. occurrent
(span vs. snap)

Continuants are wholly present (i.e., all their parts are present) at any time they are present.

Occurrents just extend in time by accumulating different temporal parts, so that , at any time they are present, they are only partially present.

Continuants are entities that are in time. Lacking temporal parts all their parts flow with them.

Occurrents are entities that happen in time. Their temporal parts are fixed in time.

Continuants can "genuinely" change in time, i.e., they can have incompatible properties at different times.

Occurrents cannot change since none of their parts keeps its identify in time.

Source: C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, L. Schneider, The WonderWeb Library of Foundational Ontologies Preliminary Report, ISTC-CNR, Italy, 2002

Critical aspects of Zachman

- Role dimension is ordinant, ordered, and purposive

- Interrogative dimension is ordinant and unordered

- Allows recursive decomposition (frameworks nested in frameworks)

- Advocates primitive model contents that facilitate complex model composition

- Abstracts time from purposive dimension

ISO/DIS 19439 – History

- CIMOSA – early "cube" framework
- CEN ENV 40 003:1990
- IFAC/IFIP Task Force on Enterprise Integration (1990 – 2002)
- GERAM:1999
- CEN TC 310 WG10 – upgrade 40 003
- ISO TC184 SC5 WG1
- Ballot closed Sept 12, 2002
- Approved with Comments (to be resolved)

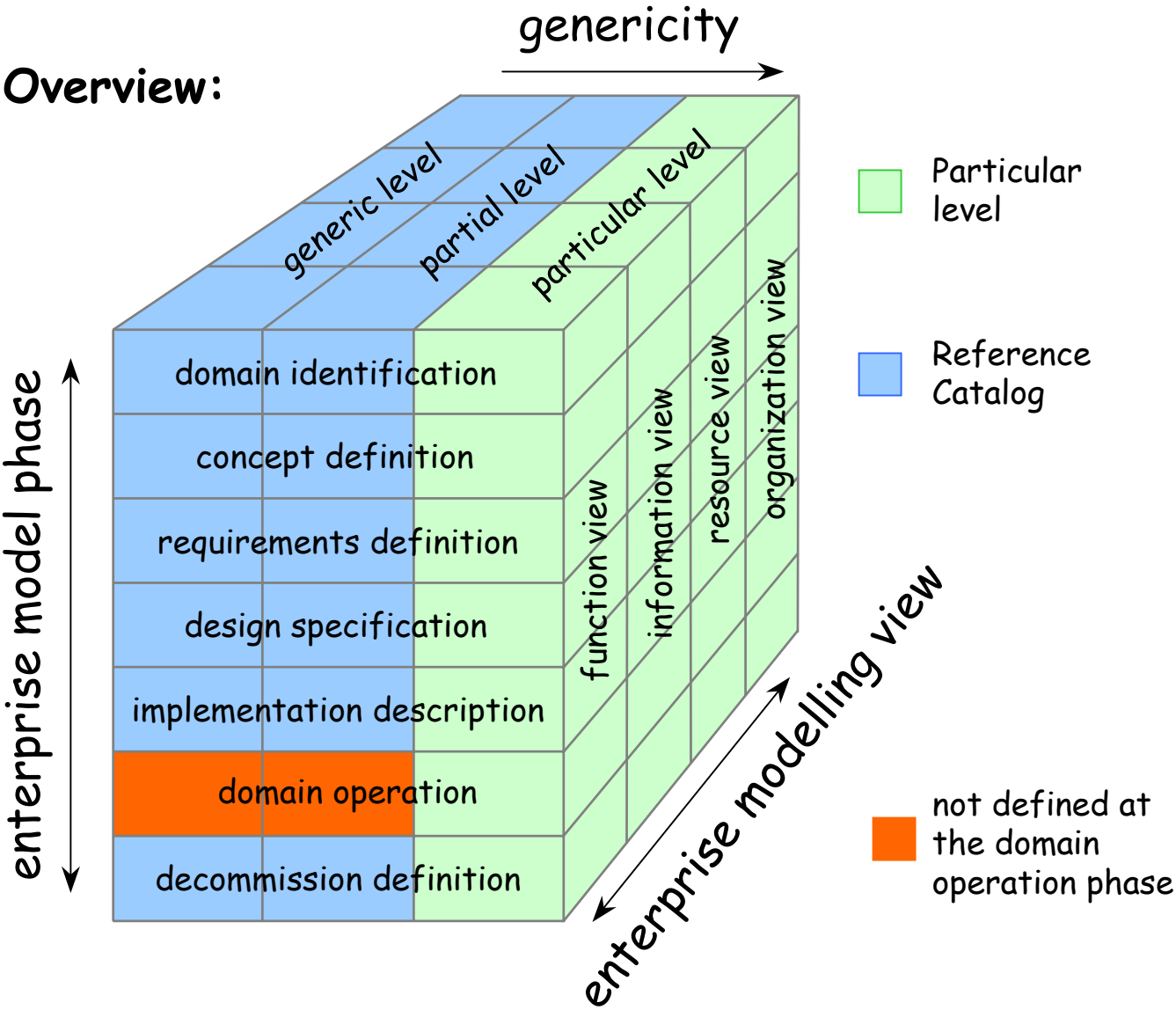
Title – Enterprise Integration – Framework for enterprise modelling

Scope – "…serves as the basis for further standards for the development of models that will be computer-enactable and enable business process model-based decision support leading to model-based operation, monitoring and control."

ISO/DIS 19439

CIM Systems Integration: Framework for Enterprise Modelling

Overview:



Source: International Standards Organization ISO/CEN parallel enquiry draft prEN ISO 19439 of 4/22/2002

ISO/DIS 19439 – Model dimension

Model – the purposive ordinant dimension ordered by coordinates corresponding to the phases of the enterprise model life-cycle.

Enterprise model phase:

- **Domain** Identification — *Identify*
- **Concept** Definition
- **Requirements** Definition
- **Design** Specification
- **Implementation** Description

  (Concept Definition … Implementation Description — *Elaborate*)
- Domain **Operation** — *Use*
- **Decommission** Definition — *Dispose*

Emphasis is on the model development process for process oriented modeling.

ISO/DIS 19439 – View dimension

View – an unordered ordinant dimension with pre-defined or user selected coordinates that emphasize aspects relevant to particular interests and context.

Enterprise modelling view:

- **Function** the system behavior, mutual dependencies, and influence of elements during function execution
- **Information** the material and information used and produced in the course of operations
- **Resource** capabilities of people and technological components
- **Organization** authority and decision-making responsibility during operations

A partitioning of facts in the integrated model

ISO/DIS 19439 – Genericity dimension

Genericity – an ordinant dimension with coordinates ordered from general to specific that reflect 19439 as a "standard" framework.

Enterprise genericity level:

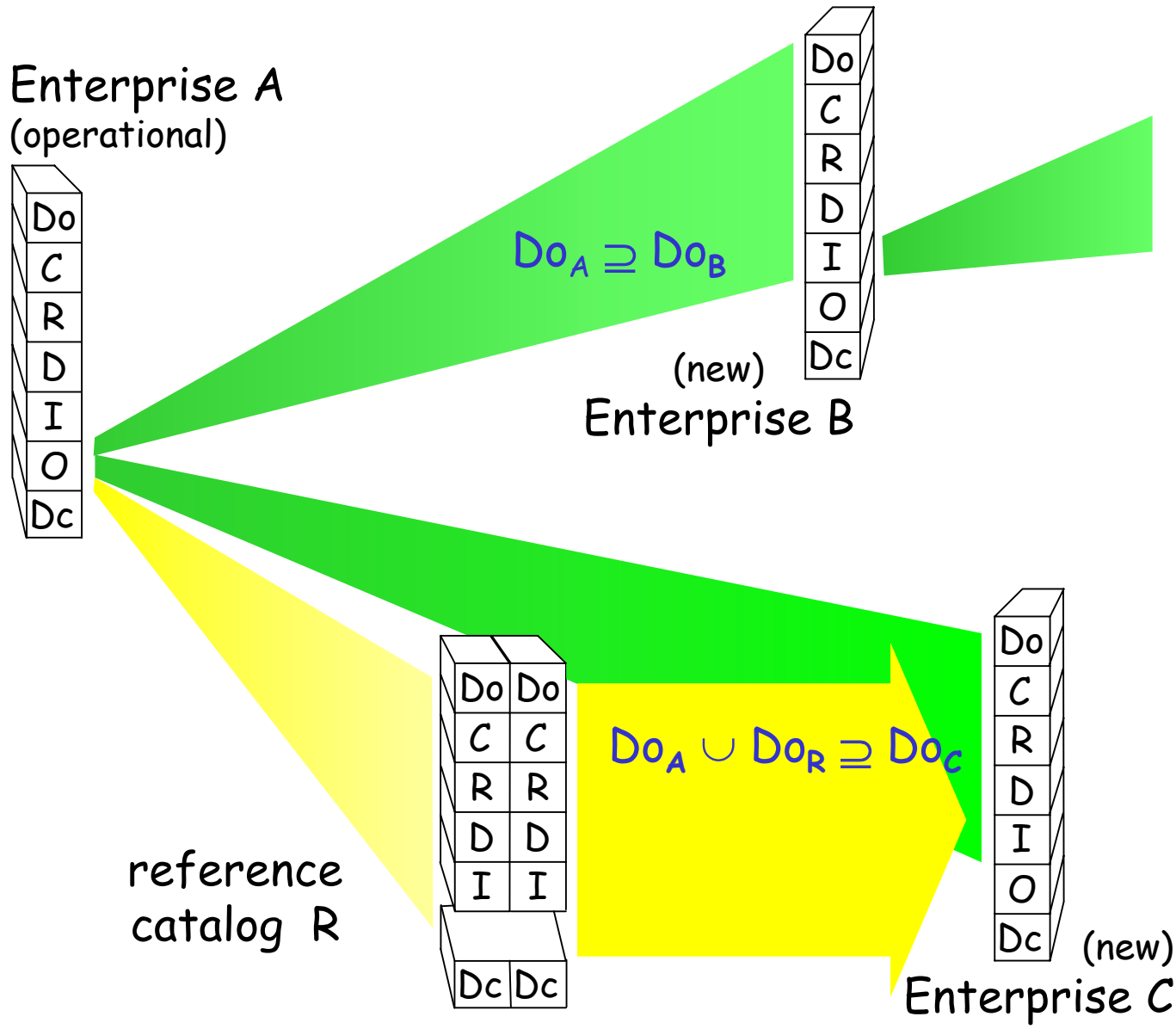
- **Generic** reusable modeling language constructs
- **Partial** prototype models of industry segment or industrial activity

Reference catalog

- **Particular** models of a particular enterprise domain

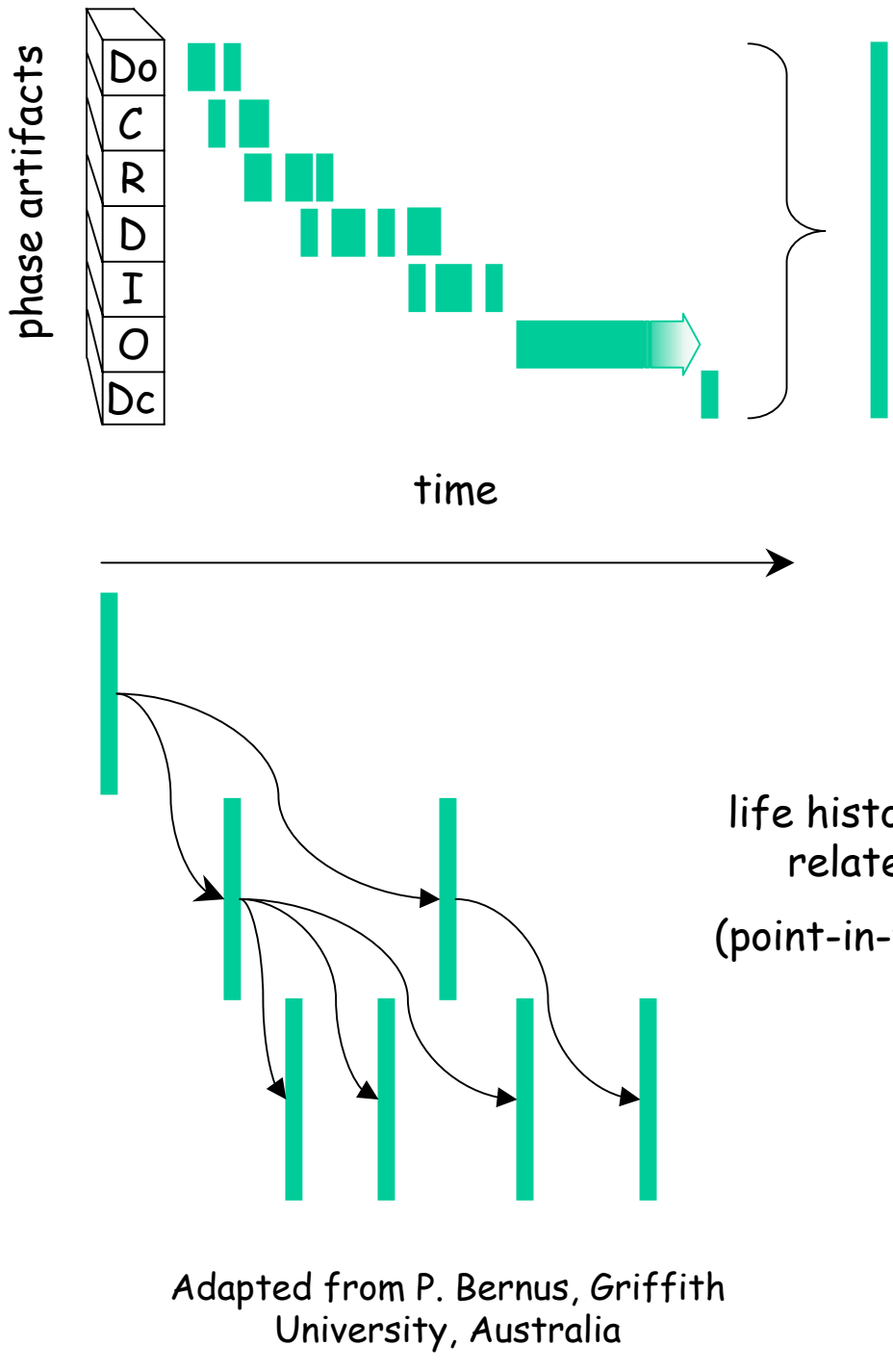
The particularization of models from the general constructs through partial models to the specialized models for an enterprise.

ISO/DIS 19439 & Recursion



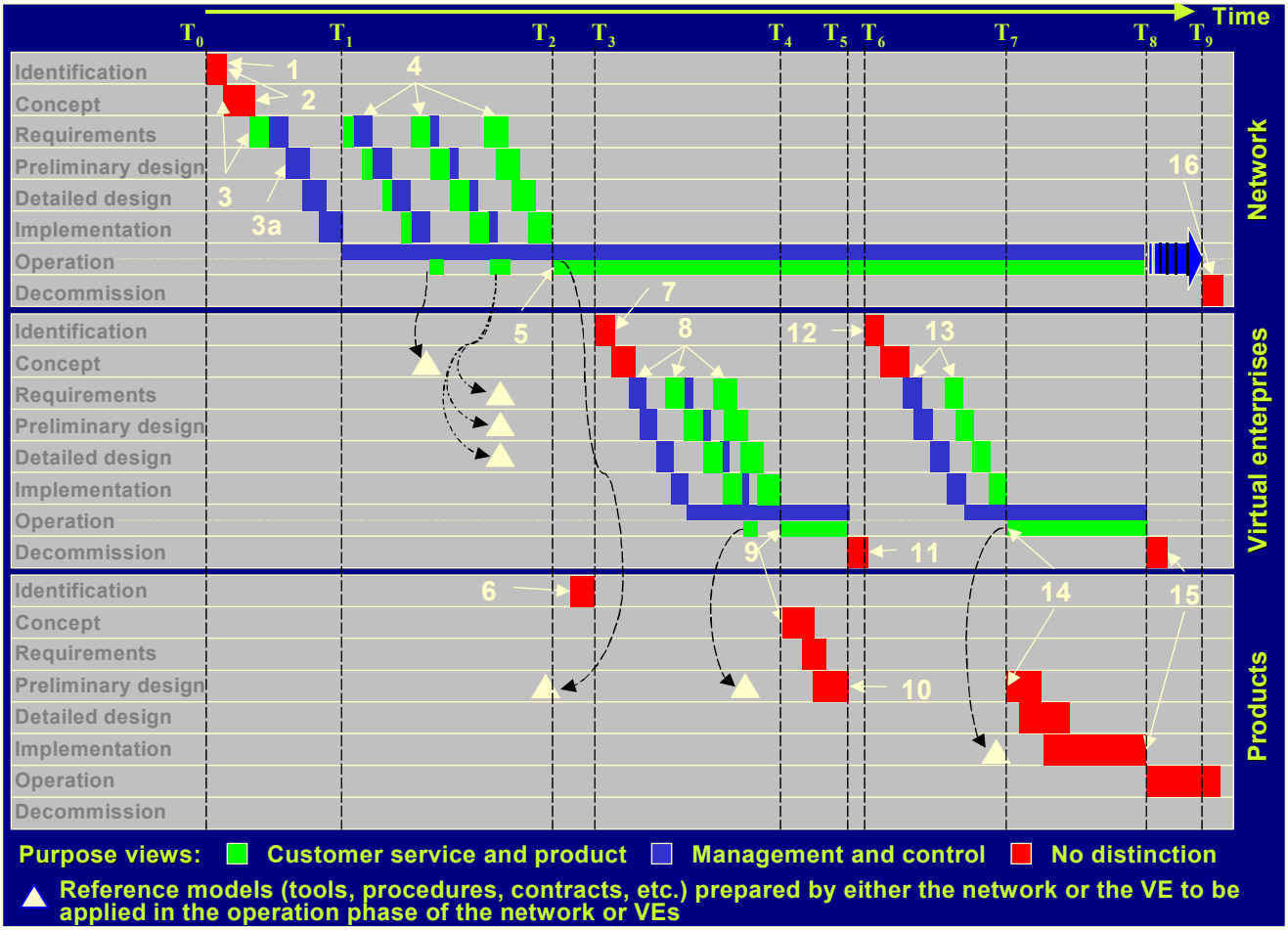
Enterprise operations can model new enterprises either from its own particular models or using reference constructs and partial models.

ISO/DIS 19439 – Life History



Adapted from P. Bernus, Griffith University, Australia

Life History example



Source: J. Vesterager, P. Bernus, J. Pedersen & M. Tolle, The what and why of a Virtual Enterprise Reference Architecture, in E-work and E-commerce: Novel solutions and practices for global networked economy. B. Stanford-Smith and E. Chiozza (Eds) IOS Press, Amsterdam (2001) Used with permission

3 GERAM instances linked by response to events and reference models

C4ISR - History

- C4ISR Integration Taskforce, Integrated Architectures Panel, 1995
- C4ISR Architecture Framework, Version 1.0 : 1996
- Clinger-Cohen Act of 1996
- C4ISR Architecture Working Group, Framework Panel, 1996
- C4ISR Architecture Framework, Version 2.0 : 1997
- C4ISR Core Architecture Data Model (CADM) Version 2.0, 1998

Title: Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework Version 2.0

Purpose: "… provides the rules, guidance, and product descriptions for developing and presenting architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures."

C4ISR Version 2.0

Architectural Views

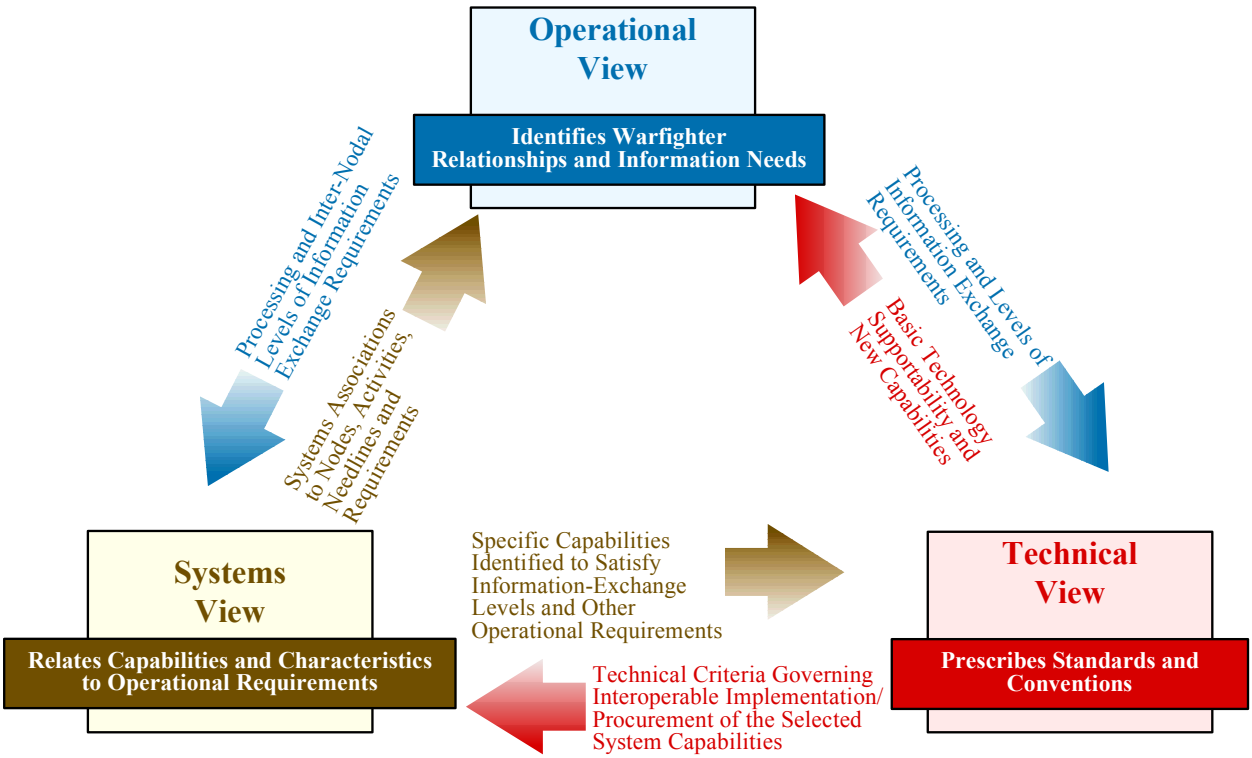


Figure 2-2. Fundamental Linkages Among the Views

"… intended to ensure that the architecture descriptions developed by the Commands, Services, and Agencies are interrelatable between and among each organization's operational, systems, and technical architecture views, and are comparable and integratable across Joint and combined organizational boundaries."

Source: Architecture Working Group, C4ISR Architecture Framework Version 2.0, 1997

C4ISR – View dimension

View – the unordered ordinant dimension with coordinates that categorize perspectives for product artifacts.

Architectural view:

- **Operational** tasks, activities, operational elements, information flows required to accomplish or support a military operation
- **Systems** the systems and interconnections providing for, or supporting, warfighting functions
- **Technical** minimal rule set for arrangement, interaction, interdependence of system parts/elements, whose purpose is ensuring a conformant system satisfies specific requirements

"…the most useful architecture description will be an 'integrated' one, i.e., one that consists of multiple views"

products may appear in more than one view

C4ISR – Guidance dimension

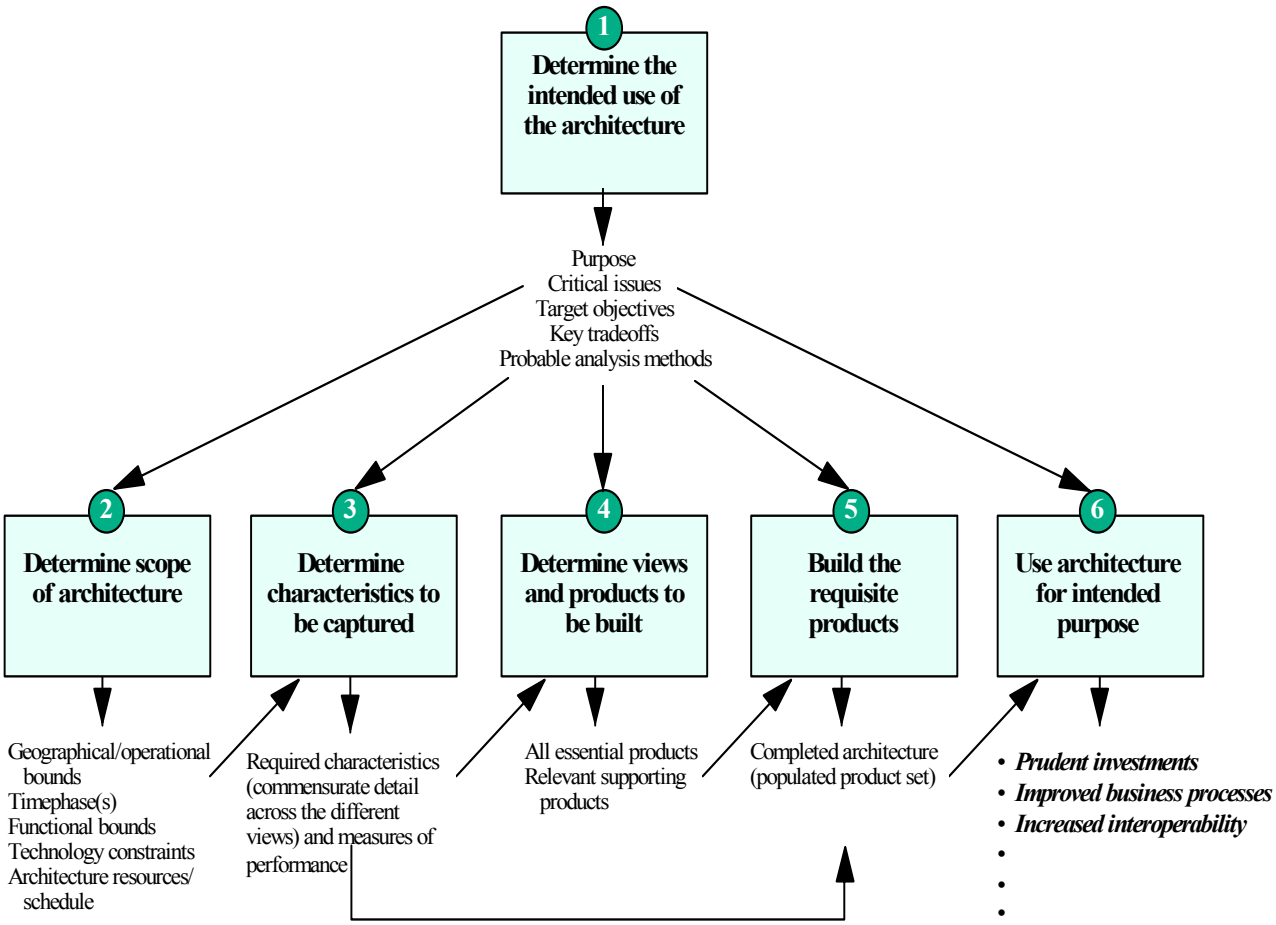


Figure 3-1. The Six-Step Process of Building an Architecture

Source: Architecture Working Group, C4ISR Architecture Framework Version 2.0, 1997

C4ISR – Guidance dimension

Guidance – a purposive ordinant dimension ordered by coordinates corresponding to the stepwise process for building an architectural product.

Description process step:

- **Focus** purpose, critical issues, objectives, trade offs, analysis methods
- **Scope** boundaries, activities, functions, organizations, timeframes, level of detail, "big picture" context, situations, areas, available resources
- **Characterize** measures of performance, extent of detail required, accommodation for future extension and use
- **Determine** views and products that portray required characteristics
- **Build** essential and requisite supporting products, consistent & properly interrelated, simulate use to test
- **Use** to enable purpose, conduct analysis

C4ISR – Integration Dimension

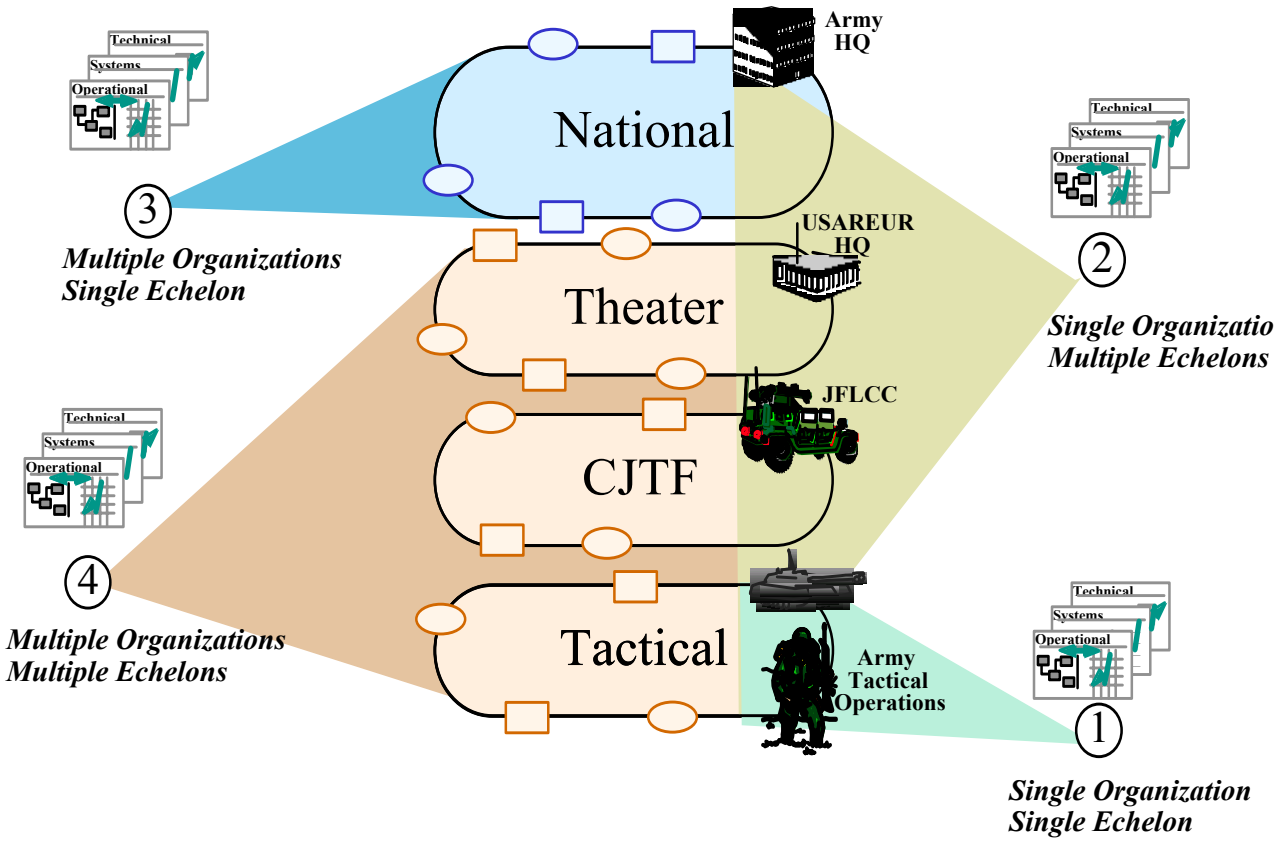


Figure 3-2. Four Dimensions of Architecture Integration

Source: Architecture Working Group, C4ISR Architecture Framework Version 2.0, 1997

C4ISR – Integration dimension

Integration – a purposive ordinant dimension ordered by coordinates corresponding to degrees of complexity in cross-architecture integration

Integration degree:

- **Multiple** organization - **multiple** echelon
  vertical and horizontal Joint relationships articulated and examined
- **Multiple** organization - **single** echelon
  horizontal unit perspectives
- **Single** organization - **multiple** echelon
  vertical operations perspectives
- **Single** organization - **single** echelon
  tactical unit perspectives

"Today, and in the near future, architecture integration will probably be accomplished toward the lower end of the integration continuum… As universal data models and standard data structures and elements emerge, integration toward the high end of the continuum will be facilitated." - C4ISR V2.0

note the embedded decomposition

C4ISR – Building Block dimension

Applicable Architecture Views	Universal Reference Resource	General Nature
All Views	***C4ISR Core Architecture Data Model (CADM)***	Logical data model of information used to describe and build architectures
All Views	***Defense Data Dictionary System (DDDS)***	Repository of standard data definitions, formats, usage, and structures
All Views	***Levels of Information Systems Interoperability (LISI)***	Reference Model of interoperability levels and operational, systems, and technical architecture associations
Operational	***Universal Joint Task List (UJTL)***	Hierarchical listing of the tasks that can be performed by a Joint military force
Operational	***Joint Operational Architecture (JOA)***	(In development) -- High-level, evolving architecture depicting Joint and multi-national operational relationships
System / Technical	***Technical Reference Model (TRM)***	Common conceptual framework and vocabulary encompassing a representation of the information system domain
System / Technical	***DII Common Operating Environment (COE)***	Framework for systems development encompassing systems architecture standards, software reuse, sharable data, interoperability and automated integration
Technical	***Shared Data Environment (SHADE)***	Strategy and mechanism for data-sharing in the context of DII COE-compliant systems
Technical	***Joint Technical Architecture (JTA)***	IT standards and guidelines

UNCLASSIFIED

Table 1. Universal Reference Resources

Source: P. K. Sowell, The C4ISR Architecture Framework: History, Status, and Plans for Evolution, 5th International Command and Control Research and Technology Symposium, 2000

C4ISR – Building Block dimension

Building block – an unordered ordinant dimension with coordinates identifying sources for terms, definitions, and specifications considered as common denominators in the DoD domain.

Building block:

- **Universal Reference Resource**
  reference models (CADM, LISI, etc.)
  information standards (DDDS, JTA, etc.)
- **Essential**
  7 C4ISR products required for high-level comparisons and budget decisions across multiple architectures
- **Supporting**
  19 C4ISR products included to achieve specific architectural intentions or characterizations
- **Additional**
  relevant to specific objectives

Essential and Supporting product interrelationships are intended to allow trace-back audit linkage.

C4ISR & Recursion

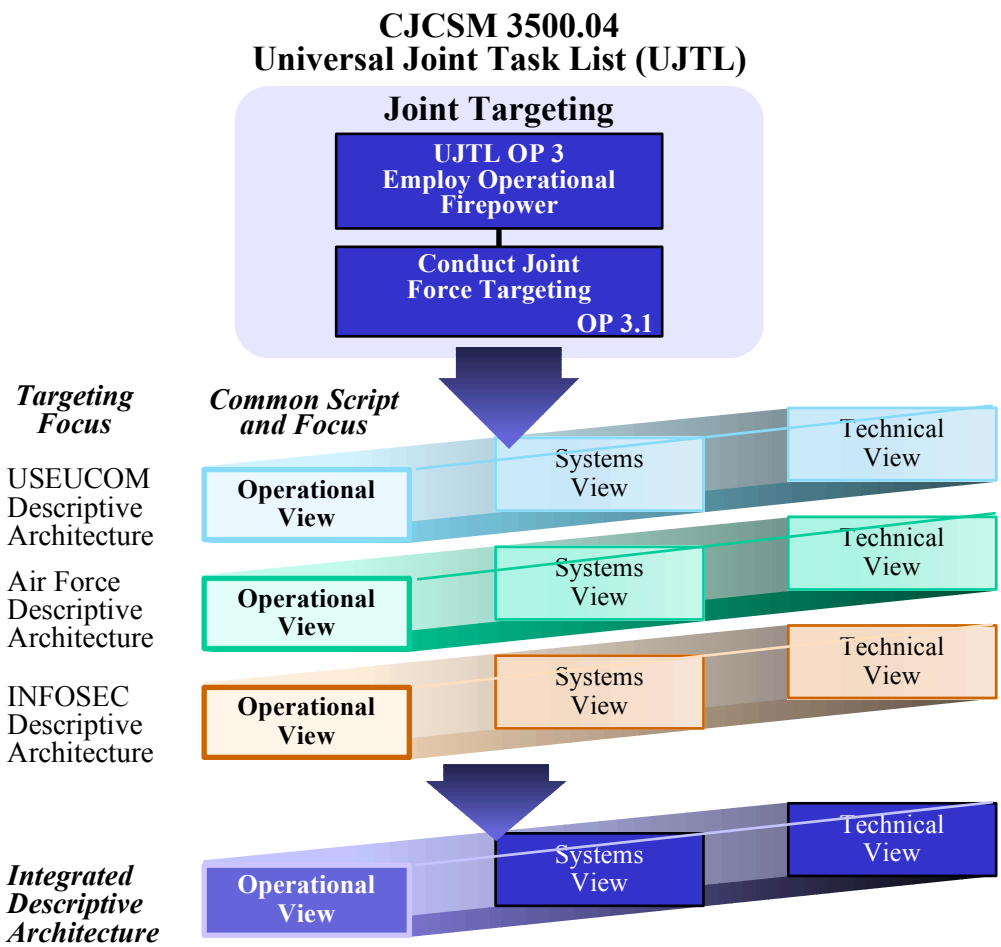


Figure 3-3. Illustration of the UJTL Serving as an Integrating Mechanism

Source: C4ISR Architecture Framework Version 2.0

the same structure at each integration level

Archetype Summary

Zachman –

Role {Context, Owner, Designer, Builder, Out-of-context}
Interrogative {What, How, Where, Who, When, Why}

ISO/DIS 19439 –

Model {Domain, Concepts, Requirements, Design, Implementation, Operation, Decommission}
View {Function, Information, Resource, Organization}
Genericity {Generic, Partial, Particular}

C4ISR –

View {Operational, System, Technical}
Guidance {Focus, Scope, Characterize, Determine, Build, Use}
Integration {Multi-Multi, Multi-Single, Single-Multi, Single-Single}
Building Block {Universal Reference Resource, Essential, Supporting, Additional}

Proto-type models

Each archetype has two levels of proto-type models:

Zachman - interrogative models {entity-relationship, input-process-output, node-link, people–work, time–cycle, ends–means}

Zachman - cell models {Semantic Model, System Design, Control Structure, Business Plan, etc.}

ISO/DIS 19439 - constructs {domain, business process, enterprise activity, event, enterprise object, resource, capability, decision centre, etc.}

ISO/DIS 19439 - partial models {industry sector, company size, national variation, etc.}

C4ISR - common terms and definitions {Core Architecture Data Model, Defense Data Dictionary System, etc.}

C4ISR - product models {High-level Operational Concept Graphic, Activity Model, Systems Rules Model, etc.}

Formal framework properties

meta-meta model	Zachman	ISO/DIS 19439	C4ISR V2.0
Structure	high	medium	low
Connections	medium	medium	high
Constraints	low	low	low
Views	low	fixed	fixed

framework descriptive terms are similar
even though model terms differ widely

Zachman is closed under composition
while ISO/DIS 19439 is not and
C4ISR has no explicit composition

Detail elaboration

These archetype frameworks have vastly different detail elaboration.

Zachman – simple column models and brief synopsis for cell content models (where's the book John?)

ISO/DIS 19439 – a construct language standard in process (ISO 19440) and EU UEML project to support more elaborate partial and particular models

C4ISR – detailed product specifications and supporting reference models with further revision as DoDAF underway

Purposive dimension

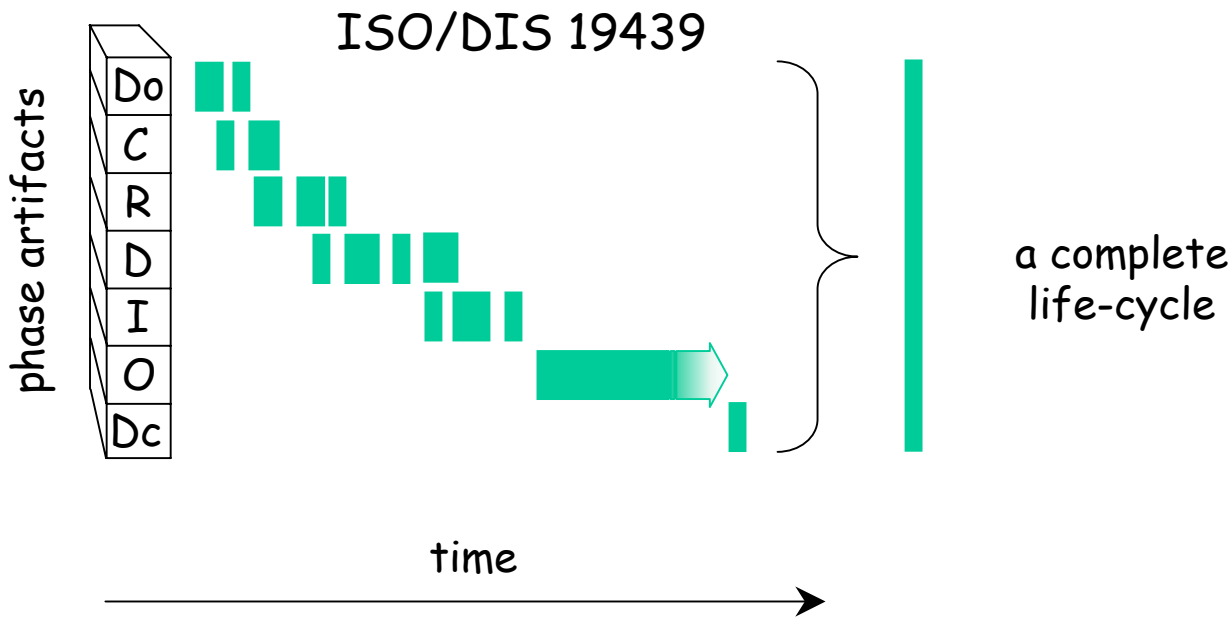
The characterization of time in the purposive dimension of a framework determines the ways in which a framework can be used.

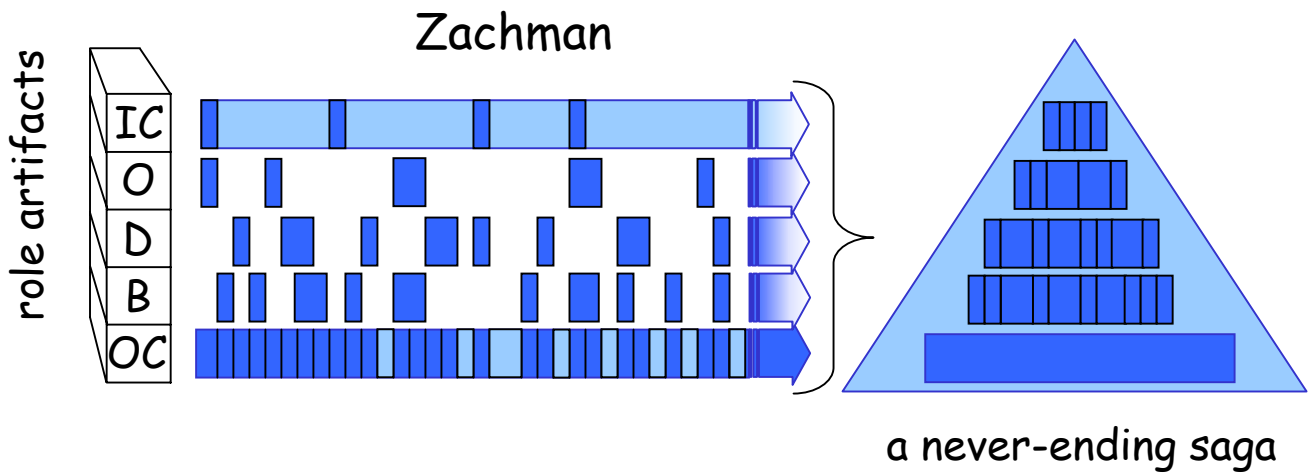
Zachman has a continuant purposive dimension (Role) and therefore serves well in an analytic resource and reference mode. It is always all there (in excruciating detail?)

ISO/DIS 19439 has an occurrent purposive dimension (Model Phase) and therefore serves well in a realization and operational mode. It provides the point-in-time solutions we use.

C4ISR has an occurrent purposive dimension (Guidance) and a continuant purposive dimension (Integration) that will result in marginal utility for the framework as the latter gains importance (an entity cannot be both continuant and occurrent).

Different Life History

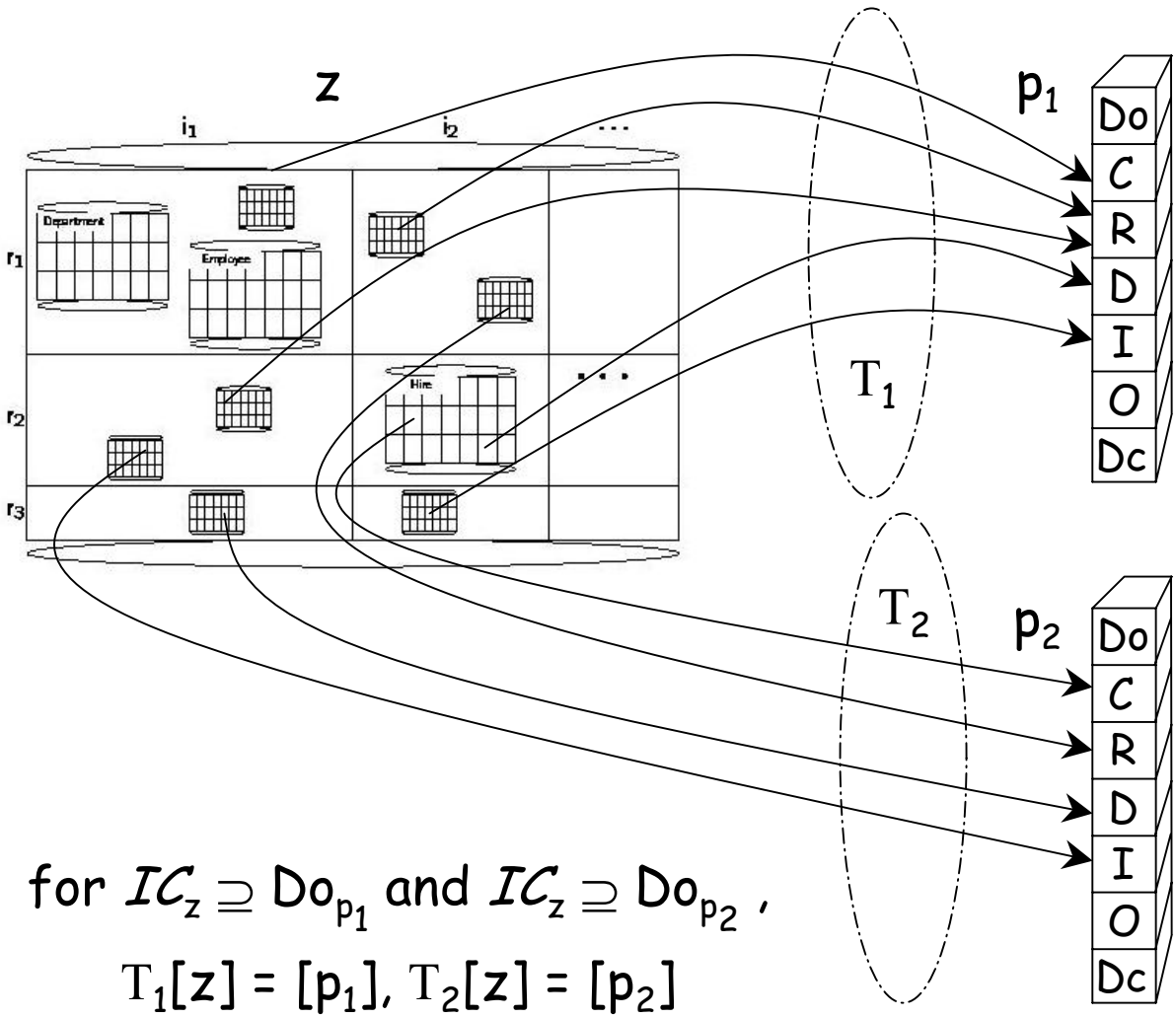




The distribution of artifact appearance in time imposes a temporal order on the purposive dimension of ISO/DIS 19439 whereas the Zachman purposive dimension order is strictly the result of dependency among artifacts.

Taking a snapshot

A Zachman continuant frame (z) can participate in an ISO/DIS 19439 occurrent frame (p). Be careful to distinguish the framework meta models from the content models.

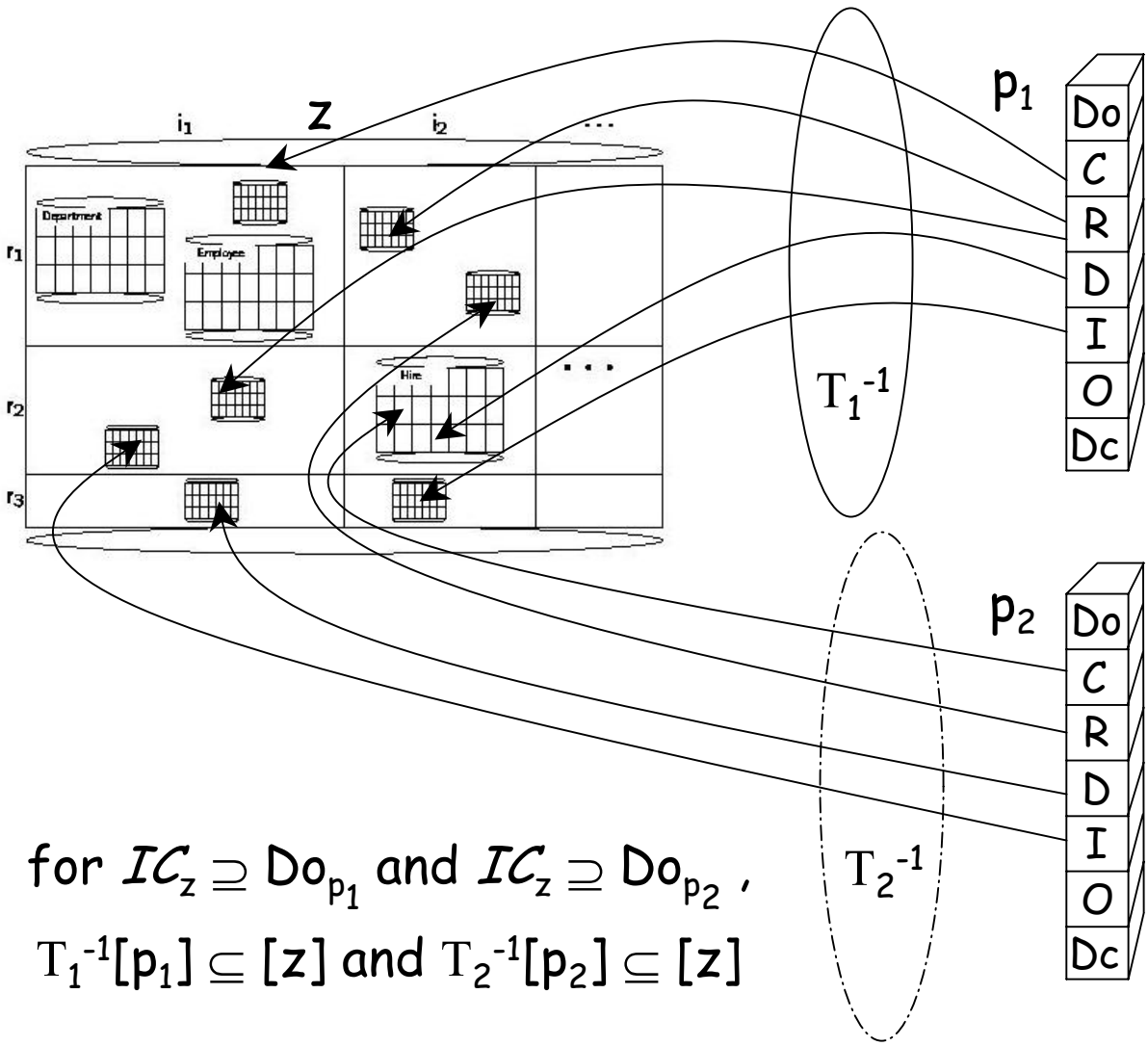


for $IC_z \supseteq Do_{p_1}$ and $IC_z \supseteq Do_{p_2}$,

$T_1[z] = [p_1]$, $T_2[z] = [p_2]$

And now the other way

An artifact produced during an occurrent realization can be rendered time neutral with respect to the framework meta model and populate a Zachman frame. That is how we get the models into a Zachman framework.



for $IC_z \supseteq Do_{p_1}$ and $IC_z \supseteq Do_{p_2}$,

$T_1^{-1}[p_1] \subseteq [z]$ and $T_2^{-1}[p_2] \subseteq [z]$

More about T and T^{-1}

Not simple inverses: T imposes a partial temporal ordering on the components of [z] used in [p] that cannot simply be withdrawn by T^{-1}

Also, let $\Pi(c)$ map an instance component into its model substructure, then for instance components c_1 and c_2,

$$\Pi(c_1) = \Pi(c_2) \Rightarrow \Pi(T(c_1)) = \Pi(T(c_2))$$

is a consistency criteria that assures a complete participation of c_1 and c_2 after T.

When p contains c_1 but not c_2 or p_1 contains c_1 and p_2 contains c_2, consistency cannot be determined.

The proper disposition of component substructure snapped across a transposition at a point-in-time will benefit from formal treatment.

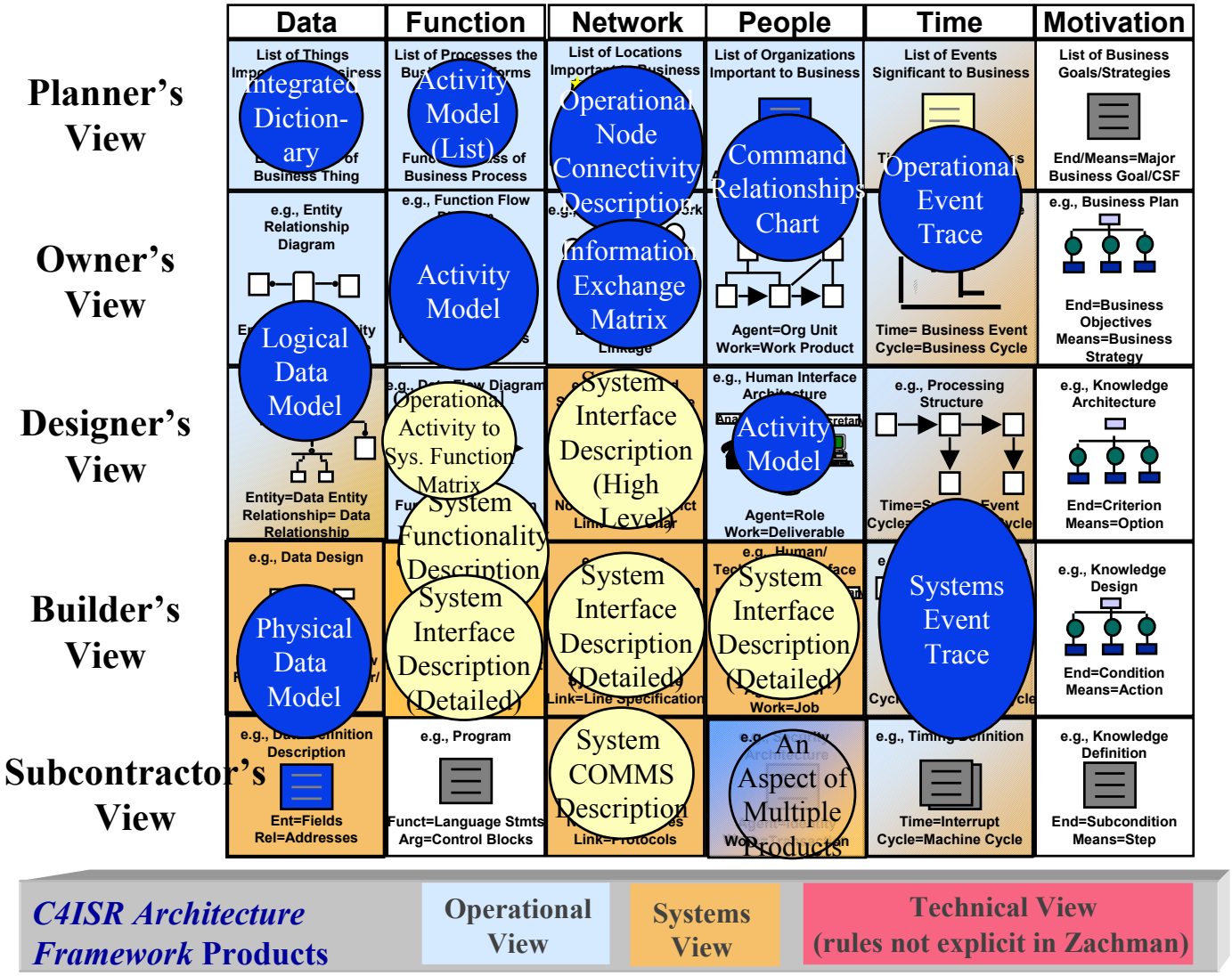
C4ISR as composite

The Guidance dimension of C4ISR is functionally equivalent to the Model dimension of ISO/DIS 19439 although with less detail. The Decommission phase is a bonus.

P. K. Sowell of Mitre has mapped most of the C4ISR products into a Zachman Framework by placing the products into appropriate cells. She has also mapped many C4ISR products into the Treasury Enterprise Architecture Framework (TEAF) with views almost identical to ISO/DIS 19439 and a purposive ordinate dimension patterned after the Zachamn role dimension.

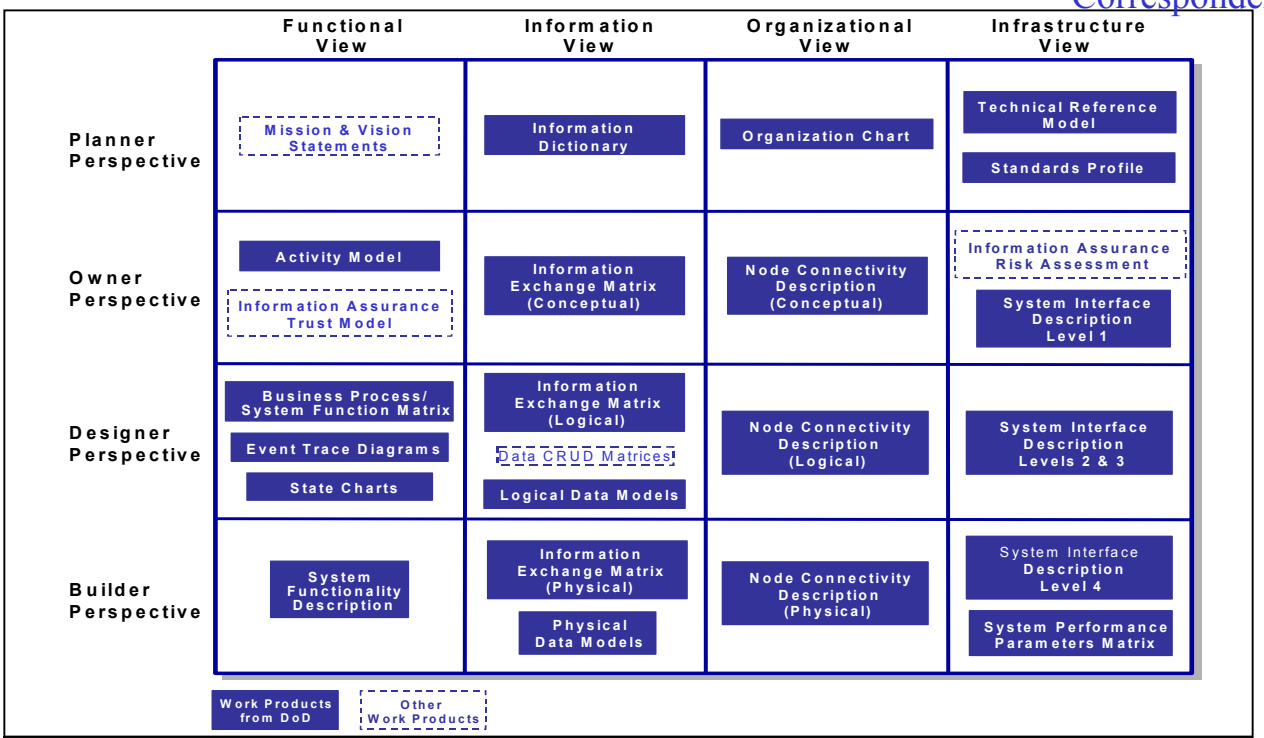
See: P. K. Sowell, The C4ISR Architecture Framework: History, Status, and Plans for Evolution, 5th International Command and Control Research and Technology Symposium, 2000

Zachman/*C4ISR Architecture Framework* Mapping



Source: P. K. Sowell, Mapping the Zachman Framework to the *C4ISR Architecture Framework*, 3 September 1999, MITRE

Other C4ISR mappings



Blue Text: C4ISR Framework Products

Shading = Products to be built in Pilot (No separate Activity Model; activities shown in Node Connectivity Description

NOTE: C4ISR Architecture Framework's "All Views" products are needed by all cells!

Perspectives	Data Architecture (entities = what)	Applications Architecture (activities = how)	Technology Architecture (locations = where)
Planner s View Objectives/ Scope	List of Business Objects	List of Business Processes *Activity Model (Hierarchy of activities*	List of Business Locations *Operational Node Connectivity -* • Major nodes only • Needlines not annotated
Owners View Enterprise Model	Semantic Model *Operational Information Exchange Matrix also contributes)*	Business Process Model	Business Logistics System *Op'nl Node Connectivity* • All nodes • Needlines annotated *Operational Information Exchange Matrix*
Designer's View Information Systems Model	Logical Data Model	Application Architecture	System Geographic Deployment *Sys. Interface Description (Internodal, System-to System)* *Technical Architecture Profile*
Builders View Technology Model	Physical Data Model	System Design	Technology Architecture
Subcontractors View Detailed Specifications	Data Definition "Library or Encyclopedia"	Programs "Supporting S/W Components"	Network Architecture

Source: P. K. Sowell, The C4ISR Architecture Framework: History, Status, and Plans for Evolution, 5th International Command and Control Research and Technology Symposium, 2000

A final observation

Suppose we have a Zachman framework, z, for our enterprise in excruciating detail and we need to respond to a change in our environment – the widget W for customer C requires a new process p. Can all that detail be used to drive the change necessary to accommodate C?

Two approaches:

$T_{W,C}[z] = [p_{W,C}]$	document the current p
$M : z \rightarrow z'$	modify z for new process
$T_{W,C}[z'] = [p'_{W,C}]$	create new process realization

or

$T_{W,C}[z] = [p_{W,C}]$	document the current p
$R_{W,C} : p_{W,C} \rightarrow p'_{W,C}$	realize new process p
$T^{-1}{}_{W,C}[p'_{W,C}] \subseteq [z']$	document new p in z

In either approach, the first step is the same. To manage change, begin with a Zachman Framework in excruciating detail!